What is Intro to Computer Science II?

Obviously, it is a continuation of CS I. But what does that mean? I would like to use the analogy to college composition I & II. In composition I, you learn how to put words together, grammatically correct, into sentences. You then learn how to put sentences together to form paragraphs. In composition II, you take those skills and learn how to create entire documents to express ideas.
For computer science, this means finding algorithmic solutions that scale well.
What is an algorithm? It is a sequence of steps to solve a problem. In a fundamental way, computer science is problem-solving. For many problems, there is not a single answer. In this course, you'll find that within broad guidelines (i.e. the requirements), much is left up to you to specify.

In CS I, you learn the symbols and syntax to create grammatically correct programs, i.e. they compile and run. You learn how to adjust the flow of control (loops and decisions) to get the programs to perform the intended work. In computer science II, you learn how to create novel solutions to problems presented to you. The assignments are more open-ended than what you had in CS I. In that course, it is important that you have structures and limits so we can ensure you are using the skills correctly. In this course, you are given requirements where it is up to you to provide the design and structure to satisfy those requirements. If you have doubts about what is required, ask!

A Big Jump

This is a big jump in 10 weeks! It is critical that you learn how to read and understand requirements. It is important that you understand how to resolve apparent gaps in the requirements to fill them in. We do not tell you how to create programs (other than requiring specific items). That also means we are not looking for "A" correct answer. If you use the required tools and the program does what is asked, then you did it (mostly) right! Enjoy your freedom!

What do I mean by freedom? The freedom comes from project and lab requirements that are intentionally non-specific. We present problems with some of the parameters required. Others are left for you. You will include

design documents for the assignments so you can outline how you resolved these gaps. You can focus on the design, and not just try to implement specific tasks provided by us. In 161, you were learning new skills and needed that structure. In 162, we try to give you a chance to think it through by yourself. We encourage you to work in groups (an important professional skill). Although all the labs and projects are individual work (except the group project), you can use the Piazza forums and your own study group for the projects and labs.

Freedom is also from student-driven learning (SDL). We encourage a similar process for you to learn new skills. You have an idea of what you need to know. We try to guess and provide study materials that we think you'll need. But what if we're wrong? SDL encourages the student to reflect on what they know. Once you determine what else you need, you can go get it. Maybe it's in a lecture. Maybe it's in the textbook. Maybe you find a video online that works. The tricky part is that what works for one may not work for another. We're here to help you find or to provide that information. This is the motivation for putting together synchronous sessions so you can work with classmates and a TA face to face (or as close as we can get in an online environment).

As with much freedom, there are responsibilities. In this case, much of the work are problems for you to solve. You must also solve the "problem" of what additional knowledge you require.

What to do?

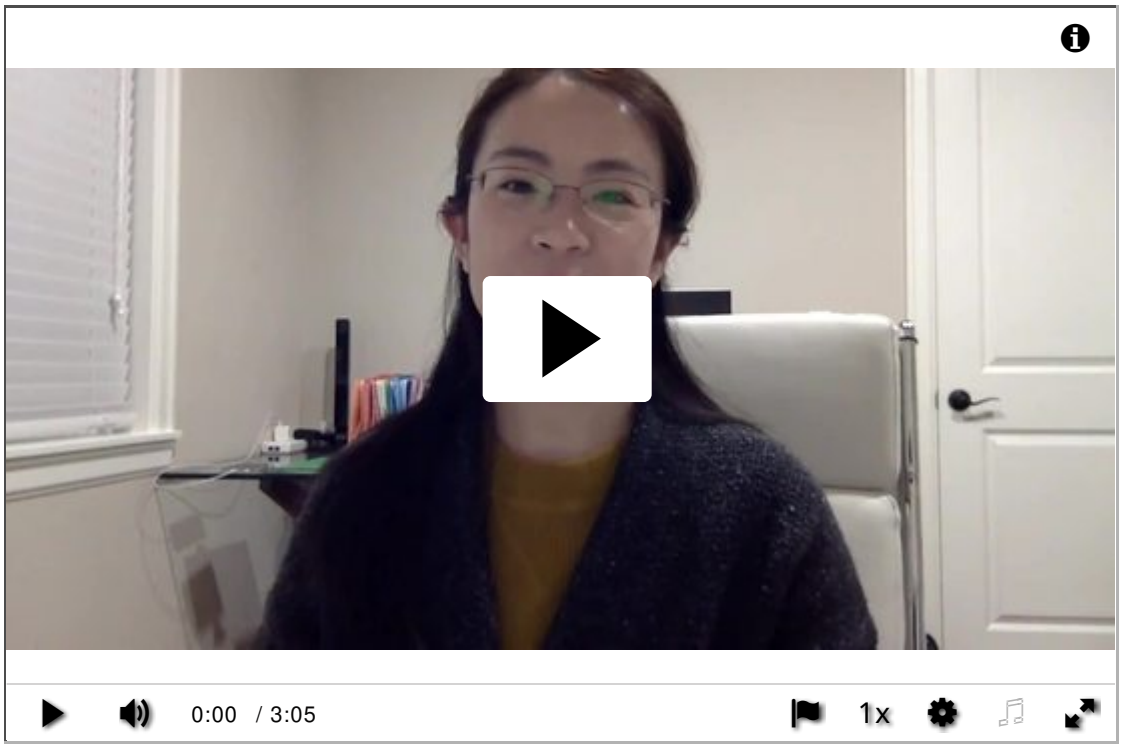
Please be aware that some links in this tab are only accessible to registered students. Please go over all the contents in the Course Syllabus page. Please read the course policy and schedule carefully. Mark the important dates like due dates of assignments and tests on your own calendar. Finish the course policy quiz and get full points to unlock week one module. There are some useful links under the resources module. Please check them out as well. You might need them for the first few assignments. Also check out the code related policy under the "Code related things" module.

Please register the course Piazza forum. There are two general discussions during the first week on Piazza. One is simply a chance for you to find classmates close to you. You should introduce yourself on CS162 Piazza forum by making a short post. We encourage you to form study groups through this discussion. The study group can be an online group or a local group where you can meet in person. The other is to start the topic discussions on Piazza. I will post more discussion topics as we introduce new contents. For more details, please check this page: A Note About Group Activities.

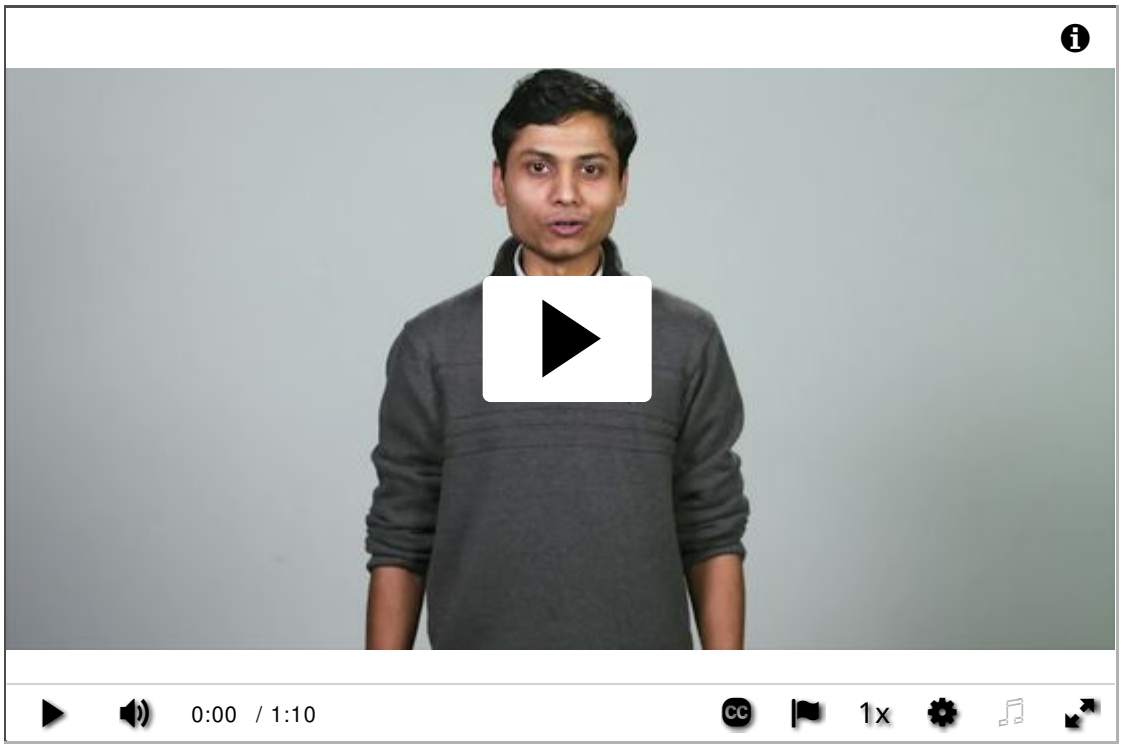
Please check the university academic calendar for start date, deadlines, holidays and so on. Also, be sure to check the "Resources" module and "Ecampus and other policies" module, where you can find information about Disabilities Access Services, The Valley Library, the OSU Computer Helpdesk, and more.

Meet the Instructors:

Dr. Luyao Zhang



Samarendra Hedaoo



Download the Course Syllabus: syllabus.pdf

CS 162: Introduction to Computer Science II

Credits: 4

Terms Offered: All terms

Instructors:

- **Luyao Zhang:** zhangluy@oregonstate.edu
- **Samarendra Hedaoo:** hedaoos@oregonstate.edu

Course Description

Basic data structures; Computer programming techniques and application of software engineering principles; Introduction to analysis of programs.

Prereqs: CS 161 or EECS 161

Textbook (required)

Gaddis et al, **Starting Out with C++: Early Objects**, 9th Edition (ISBN-13:

978-0134400242, ISBN-10: 0134400240)

Canvas

This course will be delivered via **Canvas** where you will interact with your classmates and with your instructor. Within the course Canvas site, you will access the learning materials, such as the syllabus, assignments and quizzes. For technical assistance, please visit Ecampus Technical Help.

Communication Policy

Please post all the course-related questions on **Piazza** discussion forum so that the whole class may benefit from our conversation.

Please email us for matters of a personal nature. We have two instructors in one section this term. You can contact any of us. To keep the communication more consistent, for student's last name from A to L, you can contact **Luyao Zhang** for the course related questions, and for student's last name from M to Z, you can contact **Samarendra Hedaoo.** Please include the tag **[CS162 Fall 2019]** in your email subject and use your **OSU email** instead of the Canvas email. We will reply to course-related questions and emails within 48 hours.

We will also send out the course updates through **Canvas Announcement** to the whole class and you will receive an email notification on each announcement. It is your responsibility to check your OSU email and the Canvas course website regularly to keep updated.

For questions and requests about grading and re-grading, please **post privately on Piazza and @ your grading TA**. Please don't post any grade-related request on Slack. TAs will hold office hours online (Zoom or Slack) from week 1 to week 10. The office hour schedule and link will be posted on Canvas in the "Start here" module. TA's contact information and who will grade your assignment will also be posted in the "Start here" module. These contact information will be updated within week 1.

Measurable Student Learning Outcomes

At the completion of the course, students will be able to:

1. **Design** and **implement** programs that require:

(a) **multiple classes**, structures

(b) hierarchies of classes that use **inheritance** and **polymorphism**

(c) understanding of abstraction, modularity, separation of concerns, exception handling

2. **Construct** and **use** basic **linear structures** (arrays, stacks, queues, and various linked lists) in programs, and be able to describe instances appropriate for their use.
3. **Classify** moderately complicated **algorithms** in these complexity classes: O(1), O(log n), O(n), O(n log n), and $O(n^2)$.
4. **Develop** test-data sets and testing plans for programming projects
5. **Produce recursive** algorithms, and choose appropriately between iterative and recursive algorithms.

Graded Course Works

- **Tests and Quizzes**

There will be a policies quiz at the beginning of the course, one course survey at the end of the term, and four open-book tests. Each test will have around 20 questions (T/F and multiple choices) on Canvas. You may take the test a second time, but only the score of the **final attempt** will count. Each attempt will be timed. You may not get the same questions each time.

- **Projects**

There are 4 larger programming project assignments. An assignment generally includes understanding the requirements, designing the program, implementing and testing the code, and writing a reflection document. Programs are graded on how well they solve the assigned problem, meet specifications, use proper formatting and documentation. All the 4 projects will be individual work.

- **Labs**

Labs are small development projects that reinforce the topics presented each week. You will have one lab each week, and ten labs in total. All the labs will be individual work.

- **Group Activities**

Activities include two types of activities that support the course objectives. You will discuss certain topics on some concepts and techniques on Piazza with your classmates. There is a final reflection discussion (week 9 & 10) about your journey through 161-162 to learn how to program. This discussion will be on Canvas and will be available for about 2 weeks period. There will be two project peer reviews and you need review other student's code and give comments.

- **Final Project**

There will be a final project instead of a final exam. It will be similar in the format to the projects but will cover concepts from the entire course. The final project is also individual work. It will serve as the "**Portfolio**" in this course which means you can post the entirety of the assignment publicly (e.g. Github, personal website, etc…) after the course is ended.

Grading Policies

The list below indicates how the course learning outcomes will be measured:

- Projects – 30%
- Labs – 30%
- Group activities – 10%
- Quizzes/Test – 15%
- Final project – 15%
- Total – 100%

Grade letter	Percentage floor
A	>=93
A-	>=90
B+	>=87
B	>=83
B-	>=80
C+	>=77
C	>=73
C-	>=70
D+	>=67
D	>=63
D-	>=60
F	<60

REMINDER: A passing grade for classes in CS is a **C** or above. A C- in a CS course is not considered a passing grade toward a CS degree or as a prerequisite for future CS classes.

Your grade for each assignment will be posted on **Canvas** (generally after one week of the due date). Canvas is used to simply record the scores. The final score displayed is only **approximate**. At any time, if you want a better estimate of your current grade in the course, please **email the instructor** from your OSU email account.

Re-grading: If you have a question about an assignment grade, you must contact your TA by **post privately on Piazza** within **ONE WEEK** of receiving your grade. You can use words like "regrading request lab 1 @ TA's name" in the post subject so we can see it right away. After one week, you will not be able to dispute your grade.

Late policy:

- The **final project, last lab, group activities and test MUST** be completed by the deadline, and **no late submission** would be accepted for grading purpose.
- Projects and labs should be completed by the due date. If you do not submit the assignment by the due date, there is **late penalty**:
- Late **<= 1 day**: **10%** penalty;
- **1 day** < Late **<= 3 days**: **20%** penalty;
- **3 days** < Late **<= 7 days**: **30%** penalty;
- **Late > 7 days**: **not be accepted**.

Work submitted after 7 days will not be accepted. It is your responsibility to manage your time. If there are extenuating circumstances, please contact the instructor as soon as possible **before the deadline**.

- You have **three bonus days** in the entire term to apply to any labs/project. You can use it all at once for one assignment (if you are late for 3 days), or split it and use one day each for three assignments (no "half" day).
- **How to apply the bonus day**: leave a comment on Canvas under that assignment submission, saying that you would like to apply x bonus days for this late submission, and you have y bonus days left after that. When TAs are applying late penalty, they will look at your submission time and your comments. If you don't leave a comment there, TAs will directly apply the late penalty.
- If you have a really tough situation that might affect your progress a lot (illness, job duties, family emergency...), you should contact me immediately and ask for extensions other than the 3 Bonus days. **Don't**

wait until the due date or even past the due date to explain your personal situations and ask for extensions. If you are not sure whether to ask for it, better do it.

Course Work Submission

- All work must be submitted before **23:59** (**Pacific** Time Zone) on the date they are due.
- All the files need be archived in a zip file
- Your submission must be named in the following format: assignment name + your last name + your first name for individual assignment (eg: Lab1_Jones_Adam).
- All the projects and labs must be submitted on Canvas.
- Programs must **compile and run** on the EECS server (flip) or they will not be graded. Programs must include a **makefile**.
- If you are not sure you submit the correct file, **download it and check it**. Make sure you include everything in your submission!

Topics by Weeks

Week	Topic(s)	Due
1	Pointers Review, Separate Compilation and Makefile	Lab 1 , Course policies quiz
2	Software Design, Testing and Debugging	Lab 2, Project 1,Test 1
3	Classes and Inheritance	Lab 3
4	Polymorphism and Virtual Functions	Lab 4, Project 2,Test 2, Peer review 1
5	Recursion	Lab 5
6	File I/O and Linked Lists	Lab 6, Project 3, Peer review 2, Test 3
7	ADT: Stacks and Queues	Lab 7
8	Complexity Analysis: Searching and Sorting	Lab 8, Project 4, Test 4
9	STL, Templates and Exceptions	Lab 9
10	Review of Recursion and Complexity	Lab 10, Final Reflections
Final		Final Project

You can see the assignment due dates on Canvas directly. You can also download it here: schedule.pdf

Programming assignments in this course are considered **Take Home Programming Tests**. You must do your own work, entirely.

To Do & Not To Do:

You **MAY** discuss the meaning of assignments, general approaches, and strategies with other students in the course.

You **MAY** show your code to the TAs or instructor for feedback and help.

You **MAY NOT** ask another student for help debugging your assignment code.

You **MAY NOT** use or copy code from any other source, including the internet. You **MUST** write your own code for your assignments. That doesn't mean we don't allow you to search solutions online when you meet problems. That means, if you learn how to implement some function from online recourse, do not just copy/paste the code. Make sure you cite the resources properly if the code is exactly the same, and also make sure you could write the code by yourself now without looking at the original source.

We use **plagiarism-detection software** to check your code against the code from other students. It is quite sophisticated and can easily see through **variable name changes and formatting differences**. If you are found in violation of any of the above policies, whether you are the giver or receiver of help, you will receive a zero on the assignment or fail the course (Instructor's decision). For further information, visit Academic or Scholarly Dishonesty, or contact the office of Student Conduct and Community Standards (SCCS) at 541-737-3656.

Statement Regarding Students with Disabilities

Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS), with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 541-737-4098.

NOTE for Disability Access Services (DAS) – If you have accommodations through DAS for extra time on your exams or quizzes it is very important that you communicate with your instructors as soon as possible. Ask the instructor to double check all timed exams and quizzes to make sure that extra time has

been given to you for each exam. The instructor has to do this for each timed exam or quiz manually

If you start an exam and do not see your extended time, please have your proctor call Ecampus Testing or try to contact your instructor for assistance. We can give you extra time while you are still taking the exam if we can be contacted before the exam submits.

After you read all the details in the syllabus and start here module, please take the **syllabus quiz** and get full points to unlock the week 1 module.

Instructors

Luyao Zhang zhangluy@oregonstate.edu (Office hour is held on Zoom and by appointment.)

Samarendra Hedaoo hedaoos@oregonstate.edu (Office hour is held on Zoom and to schedule one check the time slots at https://tinyurl.com/samarendra-office-hours If none of those time slots work for you, email me to schedule one.)

Please always use your OSU email to contact us and include the tag **[CS162 Fall 2019]** in the subject.

TA office hour and grading assignment information

There will be several TAs in this course and everyone will hold office hour online 2 hours per week. You will be able to visit the above page when you register this course.

We will use Piazza as our main discussion forum. Please sign up on Piazza as soon as possible.

We will also answer questions on Slack. The workspace of this course is **class-cs162-400-f19** and is available from the link on the Canvas course menu. For more information about how to use Slack, check out the above page in the start here module.

The Ecampus Student Services link includes links to academic and student service support for Ecampus students. It provides an introduction to, and information about, the OSU Computer Helpdesk, Disability Access Services, The Writing Center and Online Writing Lab, The Valley Library, and OSU Ecampus Enrollment & Student Services. Be sure to use the helpful Check Your Computer link at the top of the page to ensure you have installed the minimum

required technology for Ecampus courses.

For technical problems like VPN, software download, server access, ONID account, please use this link and contact the IT support for help.

Textbook and lecture information

The textbook we will use is "Starting out with C++: Early Objects, 9th Edition". If you bought the 8th edition, it should be fine for concept study, but the syntax in the code examples might be a little bit different since 9th edition use C++11.

In case you want to study ahead, here is the reading list of the textbook:

Chapter 5.13: Focus on Testing and Debugging

Chapter 7: Introduction to Classes and Objects (review this chapter)

Chapter 10: Pointers (review this chapter)

Chapter 11: More About Classes and Object-Oriented Programming

Chapter 15: Polymorphism and Virtual Functions

Chapter 14: Recursion

Chapter 17: Linked Lists

Chapter 18: Stacks and Queues

Chapter 9: Searching, Sorting, and Algorithm Analysis

Chapter 16: Exceptions, Templates, and STL

To successfully start this class, there are some prerequisites. Some are related to course content and outcomes from CS161. Others relate to using the Linux system on the OSU engineering server to develop programs.

Skills you must understand preferably in C++:

- Create and use classes and structs
- Create and manipulate dynamic arrays and vectors

- Create and manipulate pointers
- Create and use the following in a program:

  for loops, while loops, if statements, arrays, multidimensional arrays,

  create functions, reference parameters (call by reference)

- Allocate and deallocate dynamic memory

Requirements for our environment:

- Access the departmental servers to write, debug, compile, and execute code
- Access the Piazza forum to post questions and review answers

Other general requirements:

- Search on the internet to find solutions for problems you meet during programming
- Group discussion and team working skills